AD 414525

63-4-5

MEMORANDUM
RM-3812-PR
AUGUST 1963

QUASILINEARIZATION, SYSTEM IDENTIFICATION, AND PREDICTION

R. Bellman, H. Kagiwada and R. Kalaba

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-3812-PR
AUGUST 1963

QUASILINEARIZATION, SYSTEM IDENTIFICATION, AND PREDICTION

R. Bellman, H. Kagiwada and R. Kalaba

This research is sponsored by the United States Air Force under Project RAND—contract No. AF 49(638)-700 monitored by the Directorate of Development Planning, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA

PREFACE

An adaptive controller is one which has the capability of learning about unknown aspects of a system being controlled and then modifying its control regime in an effort to improve the quality of the control exerted.

This paper is devoted to a mathematical formulation and computational solution of the problems of system identification and the determination of unmeasurable state variables on the basis of observations of a process, two topics of central importance in the design of adaptive controllers.

The approach suggested--based on the theory of quasilinearization--is an outgrowth of continuing RAND research on the computational solution of multi-point boundary-value problems. The paper should be of interest to control engineers and numerical analysts.

SUMMARY

The dynamical equations for a system are prescribed in the form

$$\dot{x} = g(x,\alpha), \quad x(0) = c, \tag{1}$$

where the system parameter vector α and the initial vector c are unknown. Noisy observations on, e.g., the first component of the state vector, $x_1(t)$, are made at times t_i,

$$x_1(t_i) \approx b_i, \quad i = 1,2,\ldots,N \tag{2}$$

where b_i is the observation at time t_i. It is desired to find an initial vector c and a system parameter vector α which minimize the sum of the squares of the deviations

$$S = \sum_{i=1}^{N} \left\{x_1(t_i) - b_i\right\}^2. \tag{3}$$

An effective computational scheme, based on the notion of quasilinearization, is suggested. The utility of the method is illustrated by considering a process described by the Van der Pol equation, in which the system parameter and the initial velocity are to be determined on the basis of observations of the displacements at various times. A FØRTRAN program is provided.

These considerations provide a general approach to the problems of system identification and the determination of unmeasurable state variables, two matters of prime importance in the design of adaptive controllers.

CONTENTS

I. INTRODUCTION

Much of the modern theory of control processes is devoted to the minimization of functionals of the form

$$J[y] = \int_0^T h(x,y)dt \ , \tag{1}$$

where the state vector x, which is of dimension R, and the control vector y are related by the dynamical equation

$$\dot{x} = g(x,y) \ , \quad x(0) = c \ . \tag{2}$$

Though much progress has been made in the analytical and computational treatment of such problems [1,2], much remains to be done. In particular, it must be recognized that frequently it is not possible to measure all of the components of the state vector, and the dynamical equation, Eq. (2), may not be known precisely by the controller. Under these circumstances effective control is more difficult to achieve, but not impossible. Some of the progress which has been made in treating these matters is discussed in [2,3]. In addition, various questions arise concerning the choice of the index of performance in Eq. (1), and some are treated in [4].

In this Memorandum our aim is to show how many control problems involving unmeasurable state variables and unknown system dynamics lead, from the mathematical viewpoint, to nonlinear multi-point boundary value problems, and also to show how the quasilinearization technique, discussed in Refs. [5,6,7], leads to an efficient computational mode of solution. In effect, a procedure for the design of a broad class of adaptive controllers is given.

II. FORMULATION

Let us suppose that the exact dynamical equation is unknown to the controller but that its general form is known, the only unknowns being a vector of system constants α. The system equations are then assumed to be

$$\dot{x} = g(x,y,\alpha) \ . \tag{1}$$

In addition we assume that at certain times some of the components of the state vector are measured by sensing equipment and the controller is apprised of these measurements. The basic problem of feedback control is the determination of the optimal choice of the control vector y for any set of circumstances in which the controller may find itself.

We shall treat this problem by seeking to determine the values of the parameters in Eq. (1) and the solution of Eq. (1) which are in best agreement with the measurements, which reduces the problem to the classical one mentioned in Section I.

More precisely, we wish to determine the vector α and a complete set of initial conditions c so that the solution of the system

$$\dot{x} = g(x,\alpha) \ , \quad x(0) = c, \tag{2}$$

is in closest agreement with the observations. For ease of description let us suppose that observations of the first component of the state vector are available at times t_i, i=1,2,...,N. We denote the observed value at time t_i by b_i. Our aim is to determine the system parameter vector α and the initial state vector c that minimize the sum of the squares of the deviations

$$S = \sum_{i=1}^{N} \left(x_1(t_i) - b_i \right)^2, \tag{3}$$

where $x_1(t_i)$ is the first component of the vector x evaluated at $t = t_i$. As was mentioned earlier, once the minimizing system vector and initial state vector are determined, we shall consider them to be the "actual" system and initial state vectors and control the system on that basis. As the process continues, the identification and determination procedures are to be repeated from time to time, so that adaptation is possible.

III. A PRELIMINARY SIMPLIFICATION

It is inconvenient to consider the system vector α and the initial state vector c to be different types of vectors. Let us consider α to be part of the state vector, where α is subject to the equation

$$\dot{\alpha} = 0 \,. \tag{1}$$

Then our object is to determine the initial vectors $\alpha(0)$ and $x(0)$ in such a way that we minimize the sum

$$S = \sum_{i=1}^{N} (x_1(t_i) - b_i)^2 \,, \tag{2}$$

where

$$\begin{cases} \dot{x} = g(x,\alpha) \\ \dot{\alpha} = 0 \end{cases} \,. \tag{3}$$

But this is equivalent to the problem of minimizing the sum S, subject to the condition

$$\dot{x} = g(x), \quad x(0) = c \,, \tag{4}$$

where the minimization is over the initial vectors c, through a re-interpretation of $x, g(x)$, and c.

IV. QUASILINEARIZATION

Consider the sequence of vector functions $x^{(o)}(t)$, $x^{(1)}(t)$, ..., $x^{(2)}(t)$, ... defined in this recursive manner. The vector function $x^{(o)}(t)$ is chosen on the interval $0 \leq t \leq t_N$. After the function $x^{(k)}(t)$ is determined, the vector function $x^{(k)}(t)$ is taken to be the solution of the linearized system of differential equations

$$(1) \qquad \dot{x}_i^{(k+1)} = g_i(x^{(k)}) + \sum_{m=1}^{R} \frac{\partial g_i(x^{(k)})}{\partial x_m} \left(x_m^{(k+1)} - x_m^k\right),$$

$$i = 1,2,\ldots,R .$$

The initial conditions are to be selected so as to minimize the sum

$$(2) \qquad S_1 = \sum_{i=1}^{N} \left(x_1^{(k+1)}(t_i) - b_i\right)^2 .$$

Then, in many instances of practical importance, [5,7], the sequence of vectors $x^{(k)}(t)$ will converge quadratically to the desired solution of the problem of the previous section, and the vectors $x^{(k)}(0)$ will converge to the desired minimizing initial vector. Other applications of this idea to orbit determination [8] and system design [9] have been made.

V. COMPUTATIONAL CONSIDERATIONS - I

Since $x^{(k+1)}(t)$ is the solution of a linear system of differential solutions, it may be represented in the form

$$x^{(k+1)}(t) = p(t) + \sum_{i=1}^{R} c_i h^{(i)}(t) , \tag{1}$$

where the vector function $p(t)$ is the particular solution of Eq. (4.1) subject to the initial conditions

$$p(0) = 0 , \tag{2}$$

and the vector $h^{(i)}(t)$ is the solution of the homogeneous form of Eq. (4.1), the initial vector having the i^{th} row unity and the others zero. These vectors, $p(t)$ and $h^{(i)}(t)$, are all considered to be determined computationally on the interval $0 \le t \le t_N$. The scalars c_i, $i=1,2,\ldots,R$, are those that minimize the sum

$$S_1 = \sum_{j=1}^{N} \left\{ p_1(t_j) + \sum_{i=1}^{R} c_i h_1^{(i)}(t_j) - b_j \right\}^2 , \tag{3}$$

and may be determined from the normal equations [10]

$$\sum_{j=1}^{N} \left\{ p_1(t_j) + \sum_{i=1}^{R} c_i h_1^{(i)}(t_j) - b_j \right\} h_1^{(m)}(t_j) = 0 , \tag{4}$$

$$m = 1,2,\ldots,R .$$

In this way the problem of determining the optimal initial vector is made to depend upon the numerical solution of systems of linear initial-value problems and of linear algebraic equations. Let us next consider a special case.

VI. THE VAN DER POL EQUATION

Suppose that a process is described by Van der Pol's equation [11,12]

$$\begin{aligned} \dot{x} &= u \\ \dot{u} &= -\lambda(x^2 - 1)u - x \,, \end{aligned} \tag{1}$$

where now x is a scalar and λ is an unknown constant. We suppose that the following three observations of the displacement x have been made

$$\begin{aligned} x(4) &= -1.80843 \\ x(6) &= -1.63385 \\ x(8) &= -1.40456 \,. \end{aligned} \tag{2}$$

We wish to determine the unknown value of λ and both x and u for $t = 4$.

We consider the system of equations

$$\begin{aligned} \dot{x} &= u \\ \dot{u} &= -\lambda(x^2 - 1)u - x \\ \dot{\lambda} &= 0 \end{aligned} \tag{3}$$

and wish to determine $x(4)$, $u(4)$ and $\lambda(4)$ so that Eqs. (2) will be satisfied. In this instance it is not necessary to use the method of least squares since only three observations are given.

To obtain an initial approximation we observe that

$$\frac{x(6) - x(4)}{2} \approx .087 \approx \dot{x}(4) \tag{4}$$

$$(5) \qquad \frac{x(8) - x(6)}{2} \approx .114 \approx \dot{x}(6)$$

$$(6) \qquad \frac{\dot{x}(6) - \dot{x}(4)}{2} \approx .014 \approx \ddot{x}(4.0)$$

In view of Eq. (1) we are led to the initial approximation for λ,

$$(7) \qquad \lambda \approx 7 \, .$$

Next we integrate the system (3) with these initial conditions

$$(8) \qquad \begin{aligned} x(4) &= -\ 1.80843 \\ u(4) &= +\ 0.08 \\ \lambda(4) &= +\ 7.0 \end{aligned}$$

on the interval $4 \le t \le 8$ and obtain the functions $x_o(t)$, $u_o(t)$, $\lambda_o(t)$ on that interval.

To obtain the $(n+1)^{st}$ approximation, after having calculated the n^{th}, we use the linearized relations

$$(9) \qquad \dot{x}_{n+1} = u_{n+1}$$

$$\dot{u}_{n+1} = -\lambda_n\left(x_n^2 - 1\right)u_n - x_n + \left(x_{n+1} - x_n\right)\left[-2\lambda_n x_n u_n - 1\right]$$

$$+ \left(u_{n+1} - u_n\right)\left[-\lambda_n\left(x_n^2 - 1\right)\right] + \left(\lambda_{n+1} - \lambda_n\right)\left[-\left(x_n^2 - 1\right)u_n\right]$$

$$\dot{\lambda}_{n+1} = 0 \, ,$$

together with the three-point boundary conditions of Eqs. (2).

The results of a numerical experiment are summarized in the Table 1.

Table 1

THE FIRST NUMERICAL EXPERIMENT

	Initial Approx.	Iteration Two	Iteration Three	Iteration Four	True Values
x(4)	-1.80843	-1.80843	-1.80843	-1.80843	-1.8084322
u(4)	+0.08	+0.0564454	+0.0794911	+0.079366	.079366909
λ(4)	+7.0	9.91541	10.0004	10.00000	10.00000

A second experiment was carried out with poorer estimates of the initial velocity and system parameter. The results are presented in Table 2.

Table 2

THE SECOND NUMERICAL EXPERIMENT

	Initial Approx.	Iteration Two	Iteration Four	Iteration Six	True Values
x(4)	-1.80843	-1.80843	-1.80843	-1.80843	-1.8084322
u(4)	0.1000	-2.0758	0.599288	.0792063	.079366909
λ(4)	5.000	3.87992	11.4091	9.99956	10.000

A third trial in which the initial estimate of the system parameter was taken to be 20 resulted in an overflow, so that no results are available.

The data in Eq. (8) were generated by integrating Van der Pol's equation with

$$\lambda = 10.0$$

(10) $$x(o) = 1.0$$

$$\dot{x}(o) = 0.0$$

A graph is presented in Fig. 1.

The FØRTRAN program which produced these results is given next. The subroutines for integration of a system of ordinary differential equations and the numerical solution of linear algebraic equations are

> Adams-Moulton, Runge-Kutta Integration FAP Coded Subroutine (FORTRAN), Robert Causey and Werner L. Frank, Space Technology Laboratories; adapted by W. L. Sibley, The RAND Corporation, June 1961, RAND 7090 Library Routine Number X013. (See also SHARE Distribution #602.)
>
> Matrix Inversion with Accompanying Solution of Linear Equations, Burton S. Garbow, Argonne National Laboratory, February 1959, 704 SHARE Routine Number 664.

Phase II in the program is used when the number of observations equals the number of unknowns and phase III is used when the number of observations is greater than the number of unknowns.

The following is a brief summary of the FØRTRAN code for phase II. After the necessary data have been input, the initial approximation is generated by integrating the nonlinear differential equations. The (n+1)st approximation is obtained by integrating the particular and homogeneous equations, determining the unknown constants, and forming the linear combination of the particular and homogeneous solutions. This (n+1)st approximation is printed and stored as the n^{th} approximation in preparation for the next iteration. If the initial values of the (n+1)st approximation agree with those of the n^{th} approximation to 5 or more decimal places, the problem is considered solved.

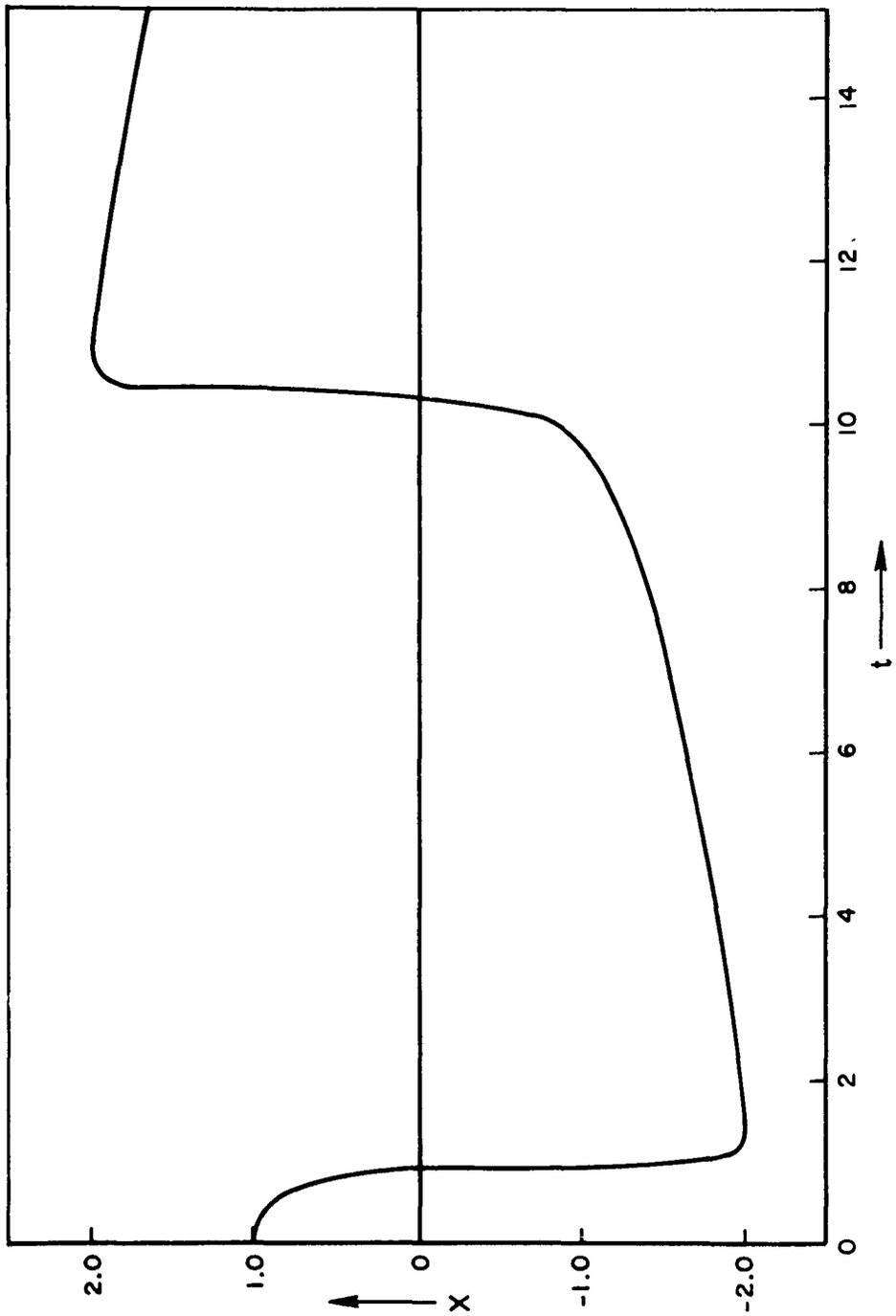

Fig. 1 — Solution of the Van Der Pol equation for $\lambda = 10$, $\dot{x}(0) = 0$, $x(0) = 1$

```
CVDP021           VAN DER POL - PHASES II, III
      COMMON SCRACH,T,N1,NMAX,KMAX,HGRID,NTIME,XOBS,W,H,P,A,B,X,U,Z,TI,
     1           PREV,NOBS,IFLAG
      DIMENSION SCRACH(64),T(150),NTIME(100),XOBS(100),W(3,1001),
     1  H(3,3,1001),P(3,1001),A(16,16),B(16,1),PREV(3)
C
C           INPUT  NOBS  OBSERVATIONS OF  X  AT KNOWN TIMES
C           DETERMINE SYSTEM PARAMETER LAMBDA (Z IN FORTRAN)
C                 INPUT AND START
   1  CALL INSTRT
C
C                 K ITERATIONS
C
      DO 99 K=1,KMAX
C
      DO 2 I=1,150
   2  T(I)=0.0
      T(2)=TI
      T(3)=HGRID
      T(4)=1.0
      T(8)=1.0
      T(12)=1.0
        X=PREV(1)
        U=PREV(2)
        Z=PREV(3)
      CALL INT(T,12,N1,0.,0.,0.,0.,0.,0.)
        N=1
        L=3
      DO 21 I=1,3
      DO 21 J=1,3
        L=L+1
  21  H(I,J,N)=T(L)
      DO 22 I=1,3
        L=L+1
  22  P(I,N)=T(L)
C
C                 INTEGRATE P'S AND H'S
      DO 4 N=2,NMAX
        X=W(1,N)
        U=W(2,N)
        Z=W(3,N)
        CALL INTM
          L=3
        DO 3 I=1,3
        DO 3 J=1,3
          L=L+1
   3    H(I,J,N)=T(L)
        DO 4 I=1,3
          L=L+1
   4    P(I,N)=T(L)
C
C                 DETERMINE CONSTANTS, OR INITIAL VALUES
      CALL CNSTNT
      TIME=TI
      PRINT 50, K,TIME,(B(I,1),I=1,3)
C
C                 NEW VARIABLES
      DO 7 N=2,NMAX
        DO 6 I=1,3
```

```
        W(I,N)=P(I,N)
        DO 6 J=1,3
    6   W(I,N)=W(I,N)+B(J,1)*H(J,I,N)
        FN=N-1
        TIME=FN*HGRID+TI
    7 PRINT 70, TIME,(W(I,N),I=1,3)
C
C                   COMPARE CONSTANTS
      DO 8 I=1,3
        G=ABSF(B(I,1)-PREV(I))
        IF(G-.000001)8,8,9
    8 CONTINUE
        GO TO 1
    9 DO 10 I=1,3
   10 PREV(I)=B(I,1)
C
   99 CONTINUE
        GO TO 1
C
   50 FORMAT(1H0//59X9HITERATION,I3//38X1HT,13X1HX,19X1HU,17X6HSYSTEM//
     1  30XF10.2,3E20.6)
   70 FORMAT(30XF10.2,3E20.6)
        END


CVDP022             INPUT-START, VAN DER POL PHASE II,III
      SUBROUTINE  INSTRT
       COMMON SCRACH,T,N1,NMAX,KMAX,HGRID,NTIME,XOBS,W,H,P,A,B,X,U,Z,TI,
     1          PREV,NOBS,IFLAG
       DIMENSION SCRACH(64),T(150),NTIME(100),XOBS(100),W(3,1001),
     1  H(3,3,1001),P(3,1001),A(16,16),B(16,1),PREV(3)
C                   INPUT
      READ 110,N1,NOBS,KMAX,NMAX,HGRID,(NTIME(I),XOBS(I),I=1,NOBS)
        IF(NMAX)9,9,1
    1 PRINT 10,N1,NOBS,KMAX,NMAX,HGRID,(NTIME(I),XOBS(I),I=1,NOBS)
C
C                   START
        IFLAG=1
      READ 120,TI,X,U,Z
        DO 2 I=1,150
    2   T(I)=0.0
        T(2)=TI
        T(3)=HGRID
        T(4)=X
        T(5)=U
        T(6)=Z
      CALL INT(T,3,N1,0.,0.,0.,0.,0.,0.)
        K=0
      PRINT 50, K,T(2),(T(I),I=4,6)
C
      DO 4 N=2,NMAX
        CALL INTM
        DO 3 I=1,3
    3   W(I,N)=T(I+3)
    4 PRINT 70, T(2),(T(I),I=4,6)
C
      IFLAG=2
      PREV(1)=X
      PREV(2)=U
      PREV(3)=Z
```

```
C
      RETURN
    9 CALL EXIT
   10 FORMAT(1H130X
     1  58HVAN DER POL - PHASE II - DETERMINATION OF SYSTEM PARAMETER//
     2  20X4I10,F10.4/(18XI12,E16.6,I12,E16.6,I12,E16.6,I12,E16.6))
   50 FORMAT(1HO//59X9HITERATION,I3//38X1HT,13X1HX,19X1HU,17X6HSYSTEM//
     1  30XF10.2,3E20.6)
   70 FORMAT(30XF10.2,3E20.6)
  110 FORMAT(4I5,F10.2,2(I4,E11.6)/(4(I4,E11.6)))
  120 FORMAT(4E12.6)
      END


CVDP023          DAUX - VAN DER POL - PHASE II,III
      SUBROUTINE DAUX
      COMMON SCRACH,T,N1,NMAX,KMAX,HGRID,NTIME,XOBS,W,H,P,A,B,X,U,Z,TI,
     1          PREV,NOBS,IFLAG
      DIMENSION SCRACH(64),T(150),NTIME(100),XOBS(100),W(3,1001),
     1  H(3,3,1001),P(3,1001),A(16,16),B(16,1),PREV(3)
C
      GO TO (1,2),IFLAG
C                NONLINEAR EQUATIONS
    1 T(7)=T(5)
      T(8)=-(T(4)**2-1.)*T(5)*T(6)-T(4)
      T(9)=0.
      RETURN
C                LINEAR EQUATIONS
    2 XX=X**2
      AA=-2.*X*U*Z-1.
      BB=Z*(1.-XX)
      CC=U*(1.-XX)
      L=13
      DO 3 I=1,4
      L=L+3
      T(L)=T(L-11)
      T(L+1)=AA*T(L-12)+BB*T(L-11)+CC*T(L-10)
    3 T(L+2)=0.0
      T(L+1)=T(L+1)+U*Z*(3.*XX-1.)
      RETURN
      END


CVDP024          CNSTNT - PHASE II
      SUBROUTINE CNSTNT
      COMMON SCRACH,T,N1,NMAX,KMAX,HGRID,NTIME,XOBS,W,H,P,A,B,X,U,Z,TI,
     1          PREV,NOBS,IFLAG
      DIMENSION SCRACH(64),T(150),NTIME(100),XOBS(100),W(3,1001),
     1  H(3,3,1001),P(3,1001),A(16,16),B(16,1),PREV(3)
C
      DO 1 I=1,3
      N=NTIME(I)
      B(I,1)=XOBS(I)-P(1,N)
      DO 1 J=1,3
    1 A(I,J)=H(J,1,N)
      CALL MATINV(A,3,B,1,DET)
      DO 2 I=1,3
    2 W(I,1)=B(I,1)
      RETURN
      END
```

VII. COMPUTATIONAL CONSIDERATIONS - II

The successive approximation method proposed, which is abstractly equivalent to Newton's method for extracting roots, shares with that procedure the desirable property of quadratic convergence. This means, roughly, that the number of correct digits is doubled with each additional iteration. Furthermore, the computational load at each stage is light, requiring only the integration of some initial-value problems and the solution of some linear algebraic equations. Nevertheless, difficulties can arise. Let us discuss some of these.

In the first place, a solution to a nonlinear multipoint boundary-value problem need not exist, nor need it be unique. With well-formulated problems arising from physical sources, however, we would not expect this to be a source of difficulty. Of more practical import is the fact that if the initial approximation is too far removed from the solution, the iterations may not converge. This possibility deserves further investigation.

We have found that the integration of the initial value problems can be difficult, for we wish to choose a grid size that is sufficiently small to give the required accuracy yet not so small as to involve excessive computing costs. Furthermore the numerical solution of the linear algebraic equations for the multipliers of the solution of the homogeneous equations can be difficult, for though the matrix involved is nonsingular, it can be ill-conditioned. A promising method for overcoming this difficulty is described in [13].

The successive approximation scheme proposed involves storing the values of the dependent variables at one stage to calculate their

values in the next. The high-speed memory limitations of the computer being used can be exceeded. A method for overcoming this is described in [14].

REFERENCES

1. Pontryagin, L. S., et al., The Mathematical Theory of Optimal Processes, Interscience Publishers, 1962, New York, N. Y.

2. Bellman, R., Adaptive Control Processes: A Guided Tour, Princeton Univ. Press, 1961, Princeton, N. J.

3. Kalaba, R., "Mathematical Aspects of Adaptive Control," a chapter in the book Mathematical Problems in the Biological Sciences, Amer. Math. Soc., 1962, Providence, R. I.

4. Bellman, R., and R. Kalaba, An Inverse Problem in Dynamic Programming and Automatic Control, The RAND Corporation, RM-3592, April 1963, Santa Monica, California.

5. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation and Monotone Convergence," Jour. of Math. and Mech., v. 8 (1959), pp. 519-574.

6. Kalaba, R., "Some Aspects of Quasilinearization," a chapter in the book Nonlinear Differential Equations and Nonlinear Mechanics, Academic Press, 1963, New York, N. Y.

7. Bellman, R., and R. Kalaba, Quasilinearization and Boundary-value Problems, Elsevier, 1963, New York, N. Y.

8. Bellman, R., H. Kagiwada, and R. Kalaba, "Orbit Determination as a Multipoint Boundary-value Problem and Quasilinearization," Proc. Nat. Acad, Sci. USA, v. 48 (1962), pp. 1327-1329.

9. Bellman, R., H. Kagiwada, and R. Kalaba, "A Computational Procedure for Optimal System Design and Utilization," v. 48 (1962), pp. 1524-1528.

10. Whittaker, E., and G. Robinson, The Calculus of Observations, Blackie and Son, 1960, London.

11. Stoker, J. J., Nonlinear Vibrations in Mechanical and Electrical Systems, Interscience, 1950, New York, N. Y.

12. Gibson, J. E., Nonlinear Automatic Control, McGraw-Hill, 1963, New York, N. Y.

13. Godunov, S. K., "On the Numerical Solution of Boundary-value Problems for Systems of Linear Ordinary Differential Equations," Uspekhi Matematicheskikh Nauk, v. 16 (1961), pp. 171-174.

14. Bellman, R., "Successive Approximations and Computer Storage Problems in Ordinary Differential Equations," Comm. Assoc. Comput. Machinery, v. 4 (1961), pp. 222-223.